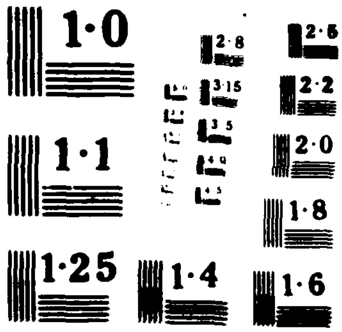
AD-A176 613 NEARLY OPTIMAL ALGORITHMS AND BOUNDS FOR MULTILAYER CHANNEL ROUTING(U) MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE.. B BERGER ET AL. DEC 86 1/1

UNCLASSIFIED VLSI-MEMO-86-351 N00014-80-C-0622 F/G 9/5 NL


1·0
2·8
2·5
3·2
2·2
3·6
1·1
4·0
2·0
1·8
1·25
1·4
1·6

(12)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY — VLSI PUBLICATIONS

VLSI Memo No. 86-351
December 1986

NEARLY OPTIMAL ALGORITHMS AND BOUNDS FOR MULTILAYER CHANNEL ROUTING

Bonnie Berger, Martin Brady, Donna Brown, and Tom Leighton

Contract N00014-80-C-0622

Abstract

This paper presents algorithms for routing channels with $L \geq 2$ layers; we not only present substantial new results, but we provide a unified framework in which many previously known results can be obtained. For the unit-vertical-overlap model, we describe a 2-layer channel routing algorithm which uses at most $d + O(\sqrt{d})$ tracks to route two-point net problems and $2d + O(\sqrt{d})$ tracks to route multipoint nets. In addition, we show that $d + \Omega(\log d)$ tracks is a lower bound on channel width for 2-layer routing allowing even arbitrary length overlap. Moreover, our algorithm can also be used to obtain the known bounds for the Manhattan and knock-knee models. We generalize the algorithm to unrestricted multilayer routing and use only $d/(L-1) + O[\sqrt{(d/L)}]$ tracks for two-point nets (within $O[\sqrt{(d/L)}]$ tracks of optimal) and $d/(L-2) + O(\sqrt{d})$ tracks for multipoint net problems (within a factor of $(L-1)/(L-2)$ times optimal).

Microsystems Research Center Room 39-321
Massachusetts Institute of Technology
Cambridge Massachusetts 02139
Telephone (617) 253-8138

Acknowledgements

This research was supported in part by the National Science Foundation under Grant no. ECS-8352185, by the Air Force Office of Scientific Research under Contract No. AFOSR-82-0326, by the Defense Advanced Research Projects Agency under Contract No. N00014-80-C-0622, by a National Science Foundation Graduate Fellowship, and by a National Science Foundation Presidential Young Investigator Award with matching funds from Xerox.

Author Information

Berger: Laboratory for Computer Science, MIT, Room NE43-328, Cambridge, MA 02139, (617) 253-6259; Brady and Brown: Department of Electrical and Computer Engineering, Campus Box 425, University of Colorado, Boulder, CO 80309 and Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801; Leighton: Department of Mathematics and Laboratory for Computer Science, MIT, Room 2-372, Cambridge, MA 02139, (617) 253-3662.

NEARLY OPTIMAL ALGORITHMS AND BOUNDS FOR MULTILAYER CHANNEL ROUTING

(preliminary draft)

Bonnie Berger[1], Martin Brady[2,3], Donna Brown[2,3], Tom Leighton[4]

[1] Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

[2] Department of Electrical and Computer Engineering
Campus Box 425
University of Colorado
Boulder, CO 80309

[3] Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801

[4] Mathematics Department and
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract. This paper presents algorithms for routing channels with $L \geq 2$ layers; we not only present substantial new results, but we provide a unified framework in which many previously known results can be obtained. For the unit-vertical-overlap model, we describe a 2-layer channel routing algorithm which uses at most $d + O(\sqrt{d})$ tracks to route two-point net problems and $2d + O(\sqrt{d})$ tracks to route multipoint nets. In addition, we show that $d + \Omega(\log d)$ tracks is a lower bound on channel width for for 2-layer routing allowing even arbitrary length overlap. Moreover, our algorithm can also be used to obtain the known bounds for the Manhattan and knock-knee models. We generalize the algorithm to unrestricted multilayer routing and use only $\frac{d}{L-1} + O(\sqrt{d/L})$ tracks for two-point nets (within $O(\sqrt{d/L})$ tracks of optimal) and $\frac{d}{L-2} + O(\sqrt{d})$ tracks for multipoint net problems (within a factor of $\frac{L-1}{L-2}$ times optimal).

This research was supported in part by NSF ECS-8352185, Air Force Contract AFOSR-82-0326, DARPA contract N00014-80-C-0622, an NSF Graduate Fellowship, and an NSF Presidential Young Investigator Award with matching funds from Xerox.

1. Introduction.

Channel routing plays a crucial role in the development of automated layout systems for integrated circuits (see, e.g., [HS71], [R82]). Most layout systems first place modules on a chip and then wire together terminals on different modules that should be electrically connected. This wiring problem is often solved by heuristically partitioning the given space into rectangular channels and then assigning to each such channel a set of wires which are to pass through it. This solution reduces a "global" wiring problem to a set of disjoint (and hopefully easier) "local" channel routing subproblems. For this reason, the channel routing problem has been intensively studied for over a decade, and numerous heuristics and approximation algorithms have been proposed ([BBL84], [BraBr85], [D76], [PL84], [RBM81], [RF82], [YK82]).

The generic form of the channel routing problem may be described as follows. The *channel* consists of a rectilinear grid of *tracks* (or rows) and *columns*. Along the top and bottom tracks are numbers called *terminals*, and terminals with the same number form a *net*. A net with r terminals is called an *r-point net*. The smallest net is a *2-point* net; if $r > 2$, we have a *multipoint* net. The channel routing problem is to connect all the terminals in each net using horizontal and vertical wires which are routed along the underlying rectilinear grid. The goal is to complete the wiring using the minimum number of tracks; i.e., to minimize the *width* of the channel.

A variety of models have been proposed for channel routing, with differences depending on the number of layers allowed and on the ways in which wires are allowed to interact. The simplest is *river routing*, in which only one layer of interconnect is available. Unfortunately, only planar problems are routable, and even a simple routable problem (like an N net shift-left-by-1) may require N tracks to river route.

Two-layer models are relevant to practice, however, and have been extensively studied. The most common 2-layer model is the traditional *Manhattan* routing model. In the Manhattan model, horizontal wire segments are routed in one layer and vertical wire segments are routed in the other layer. Hence, wires can *cross* but cannot *overlap* (i.e., run on top of one another) for any distance. Note that when a wire changes direction, it must also change layers, which requires a *contact cut* at the corresponding grid point. Where a contact cut is used, no other electrically disjoint wire can pass through that grid point in either layer.

Unfortunately, channel routing is NP-complete for many interesting routing models, including Manhattan routing and even for the special case of 2-point nets ([Sz85], [SY82]). Fortunately, however, a linear time approximation algorithm is known for Manhattan routing [BBL84] and the bounds obtained for channel width are based on the notions of *density, d*, and *flux, f*. The *density* of any channel routing problem is the maximum number of distinct nets crossing (or touching) any vertical cut of the channel. It is not difficult to see that the density of a problem is a lower bound on its channel width in the Manhattan model. A channel has flux f if f is the largest integer for which some horizontal cut spanning $2f^2$ columns splits at least $2f^2-f$ nets. (The flux definition assumes no *trivial* nets, i.e., no 2-point nets with both terminals in the same column). Flux, like density, is a lower bound on channel width ([BBL84], [BR81]) and although the flux can be as large as $\sqrt{N}$ for an N-net problem (e.g., the shift-left-by-1), it is often much smaller in practice. In [BBL84], Baker, Bhatt and Leighton devised an algorithm to route any Manhattan problem with density d and flux f in a channel of width $2d+O(f)$. For 2-point nets, the upper bound is $d+O(f)$.

The *knock-knee* model proposed by Rivest, Baratz, and Miller [RBM81] also uses two layers but does not constrain vertical wires to be routed in a different layer than horizontal wires. Hence, wires are allowed to share corners (e.g., a wire can bend on the top layer directly above a wire bending on the lower layer), but they are still not allowed to overlap. Density again serves as a lower bound on channel width in the knock-knee model. Flux does not play a role, however, and every 2-point net knock-knee problem can be routed using $2d-1$ tracks [RBM81], [BoBr82] ($4d-1$ tracks are sufficient for multipoint net problems). Somewhat surprisingly, $2d-1$ is actually optimal in the worst case; this lower bound was demonstrated by Leighton [L81] when he discovered a class of 2-point net problems which require $2d-1$ tracks for any d.

The question remained, however, whether a better algorithm could be found if a less restrictive (but still electrically sound) model were allowed. In the first part of this paper, we answer this question affirmatively by describing an algorithm that routes any 2-point net problem with density d using only $d + O(\sqrt{d})$ tracks. For multipoint net problems, the upper bound is $2d + O(\sqrt{d})$. The model used is an extension of the knock-knee model, in which wires are allowed to overlap for unit segments in the vertical direction. We call this the *unit-vertical-overlap* model.

Density is, of course, an obvious lower bound on channel width in this unit-vertical-overlap model. In addition, we present an improved lower bound for 2 layer routing. We show that some CRPs require $d + \Omega(\log d)$ tracks even when vertical overlap of arbitrary length is allowed. This is significant in that it shows that optimal routing algorithms must use $d + O(g)$ tracks, where g is some sublinear function of d.

Although we have saved only a factor of two in channel width, and even then only at the expense of weakening the model, the result is significant for several reasons. First, it is the last factor of two that can be gained; we are approaching the lower bound. Second, the factor of two is very important in practice and we are approaching algorithms that could be useful in practice. Third, the result shows that some routing problems become easier when unit-vertical-overlap is allowed. Since such overlap may be allowable in practice, the result may lead to innovations in practical routing algorithms. Fourth, and possibly most important, our algorithm presents a unified framework which can also be used to duplicate the Baker-Bhatt-Leighton results for Manhattan routing and the Rivest-Baratz-Miller results for knock-knee routing. This is somewhat surprising since the algorithms used to obtain these results were very different and highly model dependent. (Actually, our algorithm can be roughly described as a simplification and substantial generalization of the Baker-Bhatt-Leighton algorithm.) It is important to identify algorithms that are tolerant to variations in model restrictions, since real-world channel routing problems often differ from the corresponding mathematical abstractions. Hence, algorithms that are tolerant to model changes are more likely to be reducible to practice in the long run.

In the second part of this paper, we describe the extension of our algorithms to multilayer channel routing. Recent advances in fabrication technology have increased the importance of multilayer channel routing. The initial theoretical work in this area is due to Preparata and Lipski [PL84] who defined the 3-layer knock-knee model and discovered an efficient algorithm for routing any 2-point net problem in the optimal number of d tracks. For routing multipoint nets in this knock-knee 3-layer model, the best known algorithm uses $2d-1$ tracks [SP85].

Some recent results have been obtained for routing with L layers, $L \geq 3$. Hambrusch in [H85] gave an algorithm which routes 2-point nets using $\left\lceil \frac{d}{\left\lceil \frac{L-1}{2} \right\rceil} \right\rceil + 3$ tracks for $L \geq 5$ with arbitrary overlap: $\frac{2}{3}d$ and $\frac{3}{4}d$ results are obtained for $L=3$ and $L=4$, respectively. Subsequently, Brady and Brown [BraBr85] used arbitrary overlap to route any multipoint net problem using at most $\left\lceil \frac{d+2}{\left\lfloor \frac{2}{3}L \right\rfloor} \right\rceil + 5$ tracks for $L \geq 7$ (inferior results are obtained for smaller L values). For a different model, in which wires are allowed to overlap, but not on adjacent layers, their algorithm uses at most $2\frac{d}{L} + 3$ tracks, within three tracks of optimal.

Extending Leighton's lower bound of d for $L=2$ [L81], Hambrusch showed that at least $\left\lceil \frac{d}{L-1} \right\rceil$ tracks are required to route any channel routing problem with L layers, even if wires are allowed to overlap [H83]. In this paper we generalize our 2-layer algorithm in order to route any 2-point net problem using $\frac{d}{L-1} + O(\sqrt{d/L})$ tracks and using $\frac{d}{L-2} + O(\sqrt{d/L})$ to route any multipoint net problem. Notice that these results are very close to optimal in the worst case: within $O(\sqrt{d/L})$ tracks for 2-point nets and within a multiplicative factor of $\frac{L-1}{L-2}$ times optimal for multipoint nets. Both algorithms give notable improvements over previous results, and they come very close to the inherent limits imposed by the problem.

The remainder of this paper is divided into sections as follows. In Section 2, we lay the groundwork for the 2-layer unit-vertical-overlap algorithm, describing some of the basic ideas and showing how to replicate the known results for Manhattan routing and knock-knee routing. Section 3 contains the unit-vertical-overlap algorithm. In Section 4 we present improved lower bounds for overlap routing. We generalize the algorithm to multiple layers in Section 5. For simplicity, we restrict our attention to 2-point net problems in Sections 2−5; multipoint net results are discussed in Section 6. We conclude in Section 7 with some remarks and open questions.

For brevity in this extended abstract, we have omitted much of the technical material from sections 2−6; the interested reader may refer to the Appendix and the attached figures in order to provide insight into the operation of the algorithms.

2. General Strategy.

The general strategy in all of our algorithms will be to partition the channel into *blocks* of r consecutive columns. The algorithms then proceed in two phases. The first phase uses tracks in the middle section of the channel to route each net from the block in which it begins to the block in which it ends. At the end of Phase 1, all that remains is to route the nets within each block into their correct columns; Phase 2 does this using the top and bottom sections of the channel (see Figure 1).

Phase 1 is the core of the routing strategy. It proceeds a block at a time from left to right. Nets are differentiated into three types. A *vertical* net has both of its terminals in the same block. Among nets having terminals in different blocks, a *falling* net has its rightmost terminal on the bottom of the channel, while a *rising* net has its rightmost terminal on top. These definitions cover both top-to-bottom and same-side nets. While our description of the algorithms is limited to top-to-bottom nets, same-side nets are a trivial extension, and our algorithms hold for both types of 2-point nets. A *rising/falling* strategy is employed: falling (rising) nets are packed into the lowest (highest) tracks available for horizontal routing. The empty tracks between the rising and falling nets are reserved to route blocked entering nets by backtracking to the left through a *pyramid*.

The choice of r, the block size, is dependent on the routing model. For 2-layer Manhattan routing [BBL84], use a block size $r = O(f)$ to obtain a $d + O(f)$ algorithm for 2-terminal nets. For the 2-layer knock-knee model, we can take $r = 1$, and eliminate Phase 2 altogether. The resulting algorithm requires $2d-1$ tracks: d tracks for horizontal routing, each separated by an empty track to be used solely for layer changes. Thus, the knock-knee version reproduces the results of [RBM81], [BoBr82].

3. Routing in the (2-Layer) Unit-Vertical-Overlap Model.

For the unit-vertical overlap model, we choose the block size proportional to $\sqrt{d}$. Interblock routing is divided into Phases 1.1 and 1.2. Phase 1.1 routes nets between blocks in a manner very similar to the Manhattan interblock routing procedure [BBL84]. We form disjoint ending net, continuing net, and beginning net staircase patterns, but these patterns are separated vertically in this algorithm by empty tracks, rather than horizontally, as in [BBL84]. As a result, we do not need the additional columns which led to the flux term in [BBL84], but we do need additional tracks. In fact, it is assumed that upon entering a block, five empty *good* tracks, G1 - G5, have been placed among *d Manhattan routing* tracks, in the precise positions to vertically separate the staircases. These $d+5$ tracks are called *primary* tracks, and are used to route the nets between blocks in Phase 1.1.

Some of the details of Phase 1.1 may be found in Section 1 of the Appendix, and illustrated in Figure 2. Phase 1.1 assumes that five good tracks have been placed in the correct positions upon entering the current block. Phase 1.2 guarantees this invariant by positioning five empty *extra* tracks for use in the next block as good tracks. These five tracks are actually the bad tracks created in the previous block. To accomplish this, $5\left(\frac{d+10}{r}\right)$ additional empty *reserved* tracks are evenly spaced in the channel at intervals of $y = r/5$ tracks. An extra track is moved to any desired location by a series of one unit vertical jogs, as in Figure 3. The reserved tracks break up the horizontal propagation so that at most y columns are required to move any distance. Thus, a total of $5y = r$ columns is sufficient to propagate all five extra tracks. The one unit vertical jogs are done in the horizontal layer, so they do not interfere with Phase 1.1, except to cause one unit vertical overlaps.

Lemma 1: Phase 1 requires at most $(d+5) + \frac{5(d+10)}{r} + 5$ tracks.

Since Phase 1 has routed all nets to the correct blocks, the densities of the top and bottom sections of the channel are each bounded by r. For Phase 2 apply, for instance, a no-overlap, knock-knee algorithm ([RBM81], [BoBr82]) which routes a channel in $2d-1$ tracks. Thus, the top and bottom sections can each be routed using $2r-1$ tracks.

Lemma 2: Phase 2 requires at most $4r-2$ tracks.

Phases 1 and 2 together route any two terminal net channel routing problem. By Lemmas 1 and 2 we know that we have used $t = (d+5) + \frac{5(d+10)}{r} + 5 + (4r-2)$ tracks. Choosing r in order to minimize t, we see that our algorithm uses $t = d + 4\sqrt{5(d+10)} + 8$ tracks.

Theorem 1: Any two terminal net channel can be routed in two layers using $d + O(\sqrt{d})$ tracks, allowing unit-vertical-overlap.

4. Lower Bound.

Our unit-vertical-overlap algorithm is still $O(\sqrt{d})$ tracks over the obvious lower bound. However, in this section we give a better lower bound of $d + \Omega(\log d)$ tracks which holds even if vertical overlaps of arbitrary length are allowed. Technical details of the proof may be found in Section 2 of the Appendix. Define a CRP consisting of four consecutive sets of d columns each, such that the d upper terminals of set i connect to the d lower terminals of set $i+1$, for $1 \leq i \leq 3$ (see Figure 4). The actual permutation of lower terminal connections within each set is not specified. Notice that all of the nets are falling, and the density in columns $d+1$ through $3d$ is precisely d. We assume that $d+K$ tracks are available for routing.

We wish to consider only the routing within the middle two sets of the CRP. We refer to this $2d$ column subproblem as the *restricted region*. We can show that within this region, only $O(K \cdot d)$ empty unit vertical segments may exist. Since the density is d throughout the region, at most K tracks are empty at any column as well. This means that large regions exist in which no horizontal or vertical segment is empty. Routed is very restricted in such regions, and so we are able to bound the number of possible different routings within the restricted region: Only $2^{O(K \cdot d)}$ different choices are possible. But at least $d!$ different routings are required in the restricted region, corresponding to the $d!$ different permutations of the nets being routed from the upper left to lower right. This gives $2^{O(K \cdot d)} = d!$, and thus $K = \Omega(\log d)$.

Theorem 2. The optimal solution to some CRPs of density d requires $d + \Omega(\log d)$ tracks.

This lower bound proof counts only the *number* of different routings possible in a given area. We may be grossly overestimating the number of different problems that can be routed, since there are many different ways in which to route a single problem. Also, the problem we considered is not at all complex. It contains only $4d$ nets, all of them falling. Thus, it seems feasible that the lower bound could be further improved using a different strategy. We suspect that in the worst case, $d + \Omega(\sqrt{d})$ tracks may actually be required.

5. General Algorithm for L Layers.

We can extend Section 3's two layer unit-vertical-overlap routing algorithm in order to handle multilayer channel routing, in which arbitrary overlap is allowed in an L-layer channel. In our L layer algorithm, each Manhattan routing track contains up to $L-1$ different nets, in layers 2 through L. Layer 1 is reserved for vertical connections. Now, both a track and a layer (2 through L) are required to uniquely specify a net's horizontal position; each track is said to consist of $L-1$ *lanes*.

Five primary good/bad tracks are still required in order to separate staircase patterns, but now we differentiate between two types of good/bad tracks. An *empty* good/bad track contains no nets at all in any of its $L-1$ lanes, while a *partial* good/bad track contains nets in layers 3 through L (only layer 2 is empty). It is assumed that any net which exits in a given block B will be in layer 2 upon entering the block. To insure this, nets which exit in block B are swapped into layer 2 in block $B-1$, during Phase 1.2. In addition, Phase 1.2 is still used to move five empty tracks into good positions for the next block.

Notice that if two nets in the same track both exit in the same block, both may not be swapped into layer 2. Further, a net which exits in B may not be swapped into layer 2 in $B-1$ (to exit in B) if another net in its track exits in block $B-1$, and therefore is already using layer 2. To avoid these conflicts, we want to insure that at most one of the $L-1$ nets in any given primary track exits in any pair of consecutive blocks. To achieve this, group the blocks into ***sets*** of $2(L-1)$ consecutive blocks. We only require that a net be routed into the correct set of blocks in Phase 1.1. An additional step, Phase 1.0, is added to reorder the nets between blocks and to insure that two nets never exit from the same track in the same or adjacent blocks.

The Appendix provides some details of the Phase 1 routing, which uses at most $\frac{d}{L-1} + O\left(\frac{d}{L \cdot r}\right)$ tracks. Since the nets have been routed to within the correct set of $2(L-1)$ blocks in Phase 1. Thus, the top and bottom sections each have density at most $2(L-1)r$. Phase 2 employs the simple overlap routing algorithm of [BraBr85] which uses $\frac{d}{\lceil L/2 \rceil - 1}$ tracks, and so Phase 2 of our L layer algorithm uses at most $O(r)$ tracks. Phases 1 and 2 together route an L layer channel in a total of $\frac{d}{L-1} + O\left(\frac{d}{L \cdot r}\right) + O(r)$ tracks, according to Lemmas 4 and 5. The block size r, chosen to minimize this quantity, is $r = O(\sqrt{d/L})$. This implies the following theorem.

Theorem 3: Any two terminal net channel can be routed in the L layer arbitrary overlap model using $\frac{d}{L-1} + O(\sqrt{d/L})$ tracks.

6. Multipoint Nets.

Both the two-layer and the general algorithms can be extended to handle multiterminal nets. For the unit-vertical-overlap model, the strategy is to create two horizontal wires for each entering multiterminal net, one in the rising and one in the falling side of the channel. The net's falling (rising) portion ends when the rightmost block containing a lower (upper) terminal of that net is reached. This potentially doubles the number of Manhattan routing tracks required.

Theorem 4: Any multipoint channel routing problem can be routed in the (2-layer) unit-vertical-overlap model using $2d + O(\sqrt{d})$ tracks.

This same strategy applied to L layer routing would use $2d$ horizontal routing lanes and thus lead to a $\frac{2d}{L-1} + O(\sqrt{d/L})$ track algorithm. But this is inferior to even the simplest multiterminal net routing algorithm of [BraBr85]. However, if we allow two layers, 1 and L, for vertical routing, then we eliminate vertical constraints and can route with only d horizontal routing lanes.

Theorem 5: Any multipoint channel routing problem can be routed in the L layer ($L \geq 3$) arbitrary overap model using $\frac{d}{L-2} + O(\sqrt{d/L})$ tracks.

7. Remarks and Open Questions.

The work presented here has served to tie together many of the theoretical channel routing results and has resolved a number of open issues. There are two central questions left open by this work. First, is the additive $O(\sqrt{d})$ term necessary? We suspect that it is, in the worst case, even for the least restrictive 2-layer model.

Second, do multipoint net problems really require larger channel widths than 2-point net problems? Recall that very little progress had heretofore been made on this issue. For almost all models studied, the best known algorithms have in the worst case used roughly a factor of two times as many tracks for multipoint as for 2-point nets (nonadjacent overlap models being the notable exceptions). We have at least shown that this barrier breaks down for $L > 2$.

The algorithms presented in this paper are all based on the same ideas and therefore give a unified framework in which to obtain bounds for various routing models: Manhattan, knock-knee, 2 layer unit-vertical-overlap, multilayer routing with arbitrary overlap. Thus, we can hope that an improvement of a bound for any of these models might well generalize to the others. We hope in the near future to extend the universality of our method to also include known 3-layer results. In addition, we expect to be able to improve the constants in our algorithmic bounds.

8. Acknowledgements and Related Independent Work.

The unit-vertical-overlap model was first defined in unpublished work of Leighton and Pinter in the summer of 1983 at Bell Labs. At the time, Leighton and Pinter claimed (but did not publish) a $d + O(d^{2/3})$ bound on channel width for 2-point net problems (and twice that for multipoint nets). (Also in unpublished work, Brown had in 1982 independently proved a $\frac{7}{4}d$ bound for 2-point net problems, using a less demanding 2-layer model.) Subsequently, Gao proved a $\frac{3}{2}d$ upper bound for 2-point net problems using the unit-vertical-overlap model. This was later improved to $d + O(d^{2/3})$ by Hambrusch in 1985, rediscovering the original bound of Leighton and Pinter. The Hambrusch and Gao work has been combined to form [GH85]. The original works of Leighton, Pinter, and Brown have never appeared and are subsumed by this extended abstract. It is quite possible that Ron Pinter will become a coauthor of this paper, reflecting his fundamental contributions to our early results. In any case, we are indebted to him for valuable ideas and discussions.

References

[BBL84] Baker, B., S. N. Bhatt, and T. Leighton, "An Approximation Algorithm for Manhattan Routing," pp. 205-229 in *Advances in Computing Research* **2** *(VLSI Theory)*, ed

F. P. Preparata, JAI Press, Inc., Greenwich, CT (1984).

[BoBr82] Bolognesi, T. and D. J. Brown, "A Channel Routing Algorithm with Bounded Wire Length," unpublished manuscript, Coordinated Science Lab., University of Illinois at Urbana-Champaign (1982).

[BraBr85] Brady, M. and D. J. Brown, "Optimal Multilayer Channel Routing With Overlap," *4th MIT Conf. on Advanced Research in VLSI*, pp. 281-296 (Apr. 1986).

[BR81] Brown, D. J. and R. L. Rivest, "New Lower Bound for Channel Routing," *Proc. 1981 CMU Conf. on VLSI*, pp. 178-185 (Oct. 1981).

[D76] Deutsch, D. N., "A Dogleg Channel Router," *Proc. 13th Design Automation Conference*, pp. 425-433 (June 1976).

[GH85] Gao, S. and S. Hambrusch, *Two-Layer Channel Routing with Vertical Unit-Length Overlap*, draft, March 1985.

[H83] Hambrusch, S., "Using Overlap and Minimizing Contact Points in Channel Routing," *Proc. 21st Annual Allerton Conf. on Comm., Control, and Comp.*, pp. 256-257 (Oct. 1983).

[H85] Hambrusch, S., "Channel Routing Algorithms for Overlap Models," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* **CAD-4**(1) pp. 23-30 (Jan. 1985).

[HS71] Hashimoto, A. and J. Stevens, "Wire Routing by Optimizing Channel Assignment Within Large Apertures," *Proc. 8th IEEE Design Automation Workshop*, pp. 214-224 (1971).

[L81] Leighton, F. T., "New Lower Bounds for Channel Routing," MIT VLSI Memo 82-71 (1981).

[PL84] Preparata, F. P. and W. Lipski, Jr., "Optimal Three-Layer Channel Routing," *IEEE Trans. on Computers* **C-33** pp. 427-437 (1984).

[RBM81] Rivest, R. L., A. Baratz, and G. Miller, "Provably Good Channel Routing Algorithms," *Proc. 1981 CMU Conf. on VLSI Systems and Computations*, pp. 153-159 (Oct. 1981).

[R82] Rivest, R. L., (Placement and Interconnect) System"" "The "PI" (Placement and Interconnect) System," *Proc. 19th IEEE Design Automation Conference*, (1982).

[RF82] Rivest, R. L. and C. M. Fiduccia, "A Greedy Channel Router," *Proc. 19th Design Automation Conference*, pp. 418-424 (June 1982).

[SP85] Sarrafzadeh, M. and F. P. Preparata, "Compact Channel Routing of Multiterminal Nets," *Annals of Discrete Math.* **109**North Holland, (Apr. 1985).

[SY82] Szymanski, T. G. and M. Yannakakis, personal communication 1982.

[Sz85] Szymanski, T. G., "Dogleg Channel Routing is NP-Complete," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* **CAD-4**(1) pp. 31-41 (Jan. 1985).

[YK82] Yoshimura, T. and E. S. Kuh, "Efficient Algorithms for Channel Routing," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* **CAD-1**(1) pp. 25-35 (Jan. 1982).

APPENDIX

Some Technical Details

1. Phase 1.1 for the 2-Layer Unit-Vertical-Overlap Model.

Phase 1.1 consists of the following six steps. We refer the reader to the example worked through in Figure 2.

Step 1: Nets having their rightmost terminal in the current block B (and the other terminal not in B) are called *ending* nets. Ending falling (rising) nets exit in the leftmost columns, ordered from the highest to lowest (lowest to highest) (Figure 2a).

Step 2: Nets having one terminal in a block to the left of B and the other in a block to the right of B are called *continuing* nets. Tracks vacated in Step 1 are filled in using the highest falling (lowest rising) continuing nets, using G5 (G1) to separate the staircase patterns (Figure 2a). Each good track used results in an empty track at the right of the block, called a *bad* track. B5 in Figure 2a is the bad track corresponding to G5.

Step 3: If there are more ending rising (falling) nets than ending falling (rising) nets, balance the difference by routing *starting* rising (falling) nets (i.e., nets with their leftmost terminal in B). Then, expand the pyramid into the vacated tracks. Notice that two more good tracks, G2 and G4, are used in this step (Figure 2b).

Step 4: If the remaining number of starting falling nets is greater than (less than) the number of starting rising nets, route the starting falling (rising) nets directly, as in Figure 2c.

Step 5: Route the remaining starting rising (falling) nets through the pyramid, separated from the starting falling (rising) nets by G3 (Figure 2c).

Step 6: Conclude the routing of block B by routing all vertical nets trivially across the channel.

2. 2-Layer Lower Bound.

We discuss here some details of the lower bound proof within the restricted region. Consider a horizontal (vertical) cut between two tracks (columns). A net with a terminal on either side of the cut must use one or more unit-vertical (horizontal) segments to cross the cut. The first segment used, traveling downward from upper to lower terminals is called an *initial crossing*. Our counting argument hinges on initial crossings; unit segments containing non-initial crossings are treated just as unused, or *empty* segments.

We make the following observations about routing within this region:

1. There are at most K unit-vertical-overlaps in which both wires are initial crossings in any column. This is because each unit overlap causes an empty track on either side of the column.
2. Exactly d horizontal segments per column contain initial horizontal crossings, since the density is d at all columns.
3. At most $O(K \cdot d)$ vertical segments may fail to contain an initial crossing. The set 1 and set 3 nets each cause at least $1+2+3+ \cdots +d = \frac{(d+1)d}{2}$ initial crossings. At most K set 2 nets can leave the restricted region, so they add at least $(d-K)(d+K)$ initial crossings, and from Observation 1, only $2dK$ initial crossings can share a single vertical segment.

Our initial strategy is to bound the number of different choices allowed for routing a single column. Initially, the order in which set 1 nets enter the leftmost column will be decided by the choice of routings made within the restricted region, so that the correct connections are made to the lower left terminals.

There are $2^{O(d \cdot K)}$ ways to divide the $O(d \cdot K)$ empty vertical segments between the $2d$ columns. Define each column i to have k_i empty segments, such that $\sum_{i=1}^{2d} k_i = O(d \cdot K)$. The column can be divided into $K+1$ groups of consecutive initial horizontal crossings, separated by K empty horizontal segments. Routing within a group is restricted in the following sense. The highest net which turns downward determines all routing below it in a group. This claim is illustrated by the examples in Figure 5. Further, all vertical segments in the group above the highest downward turning net obviously contain no vertical initial crossings. But since there are at most k_i "empty" vertical segments, the highest dropping net must have been within k_i tracks of the top of the group. There are $O(2^K \cdot 2^{k_i})$ ways to divide the k_i empty segments between the $K+1$ groups. This done, there remain only $O(k_i)$ tracks whose routing is not yet determined, and so in all, the number of possible routings for column i is bounded by $2^{O(k_i+K)}$.

Now consider the entire region of $2d$ columns. The maximum number of different routings for the entire region is $\prod_{i=1}^{2d} 2^{O(k_i+K)} = 2^{O(Kd)}$, since $\sum_{i=1}^{2d} k_i \le 2Kd$. But we must be able to route at least $d!$ different problems in the region, corresponding to $d!$ different permutations for the nets of set 2. Thus, $2^{O(Kd)} = d! > \left(\frac{d}{e}\right)^d$, and so $K = \Omega(\log\ d)$.

3. L Layer Algorithm.

Phase 1.0 uses a greedy strategy to reorder nets in each set of $2(L-1)$ blocks. Nets are selected for a given block just before it is to be routed. An exiting net in track t is a feasible selection for block B if no other net in t has been selected to exit in B or $B-1$. From the feasible set, simply select a net from the track containing the most nets which exit in this set. Then update the feasible set, and repeat until no further additions to the current block are possible. The remaining terminals are filled by starting and vertical nets, in any order.

Phase 1.1 is organized just as in the unit-vertical-overlap algorithm into six steps. Phase 1.1 routing for L layers is illustrated in Figure 6. There are now two types of interlocking staircases. Ending nets exit only from layer 2, and so G1 and G5 need not be completely empty. G1 and G5 are *partial* good tracks, having only layer 2 unoccupied. On the other hand, good tracks must be completely empty when continuing nets are used to fill in the vacated lanes, so that falling (rising) nets remain packed into the bottommost (topmost) primary tracks. The pyramid is formed in all $L-1$ horizontal layers as well, so G2, G3, and G4, called *empty* good tracks, are completely empty in order to separate all $L-1$ lanes of the corresponding staircases.

Phase 1.2 has two purposes in the L layer algorithm. First, the five extra tracks are moved into good track positions for the next block. Second, all nets which exit in the next block are swapped into layer 2. The reserved tracks are evenly spaced $y = r/8$ tracks apart in order to achieve both of these requirements.

The first $5y$ columns of the block are used to move five extra tracks (two partial, three empty) into good track positions for the next block, exactly as in the previous algorithm. The remaining $3y$ columns are used to swap exiting nets into layer 2. This is done by propagating another empty extra track through the entire width of the channel. Any track containing a net to be swapped into layer 2 will be jogged and swapped using three columns, as shown in Figure 7.

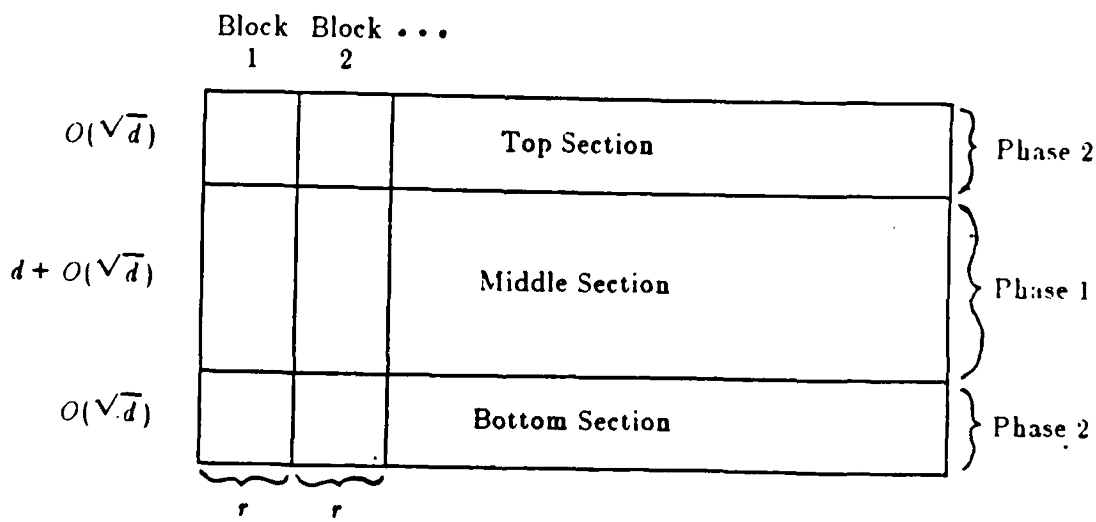
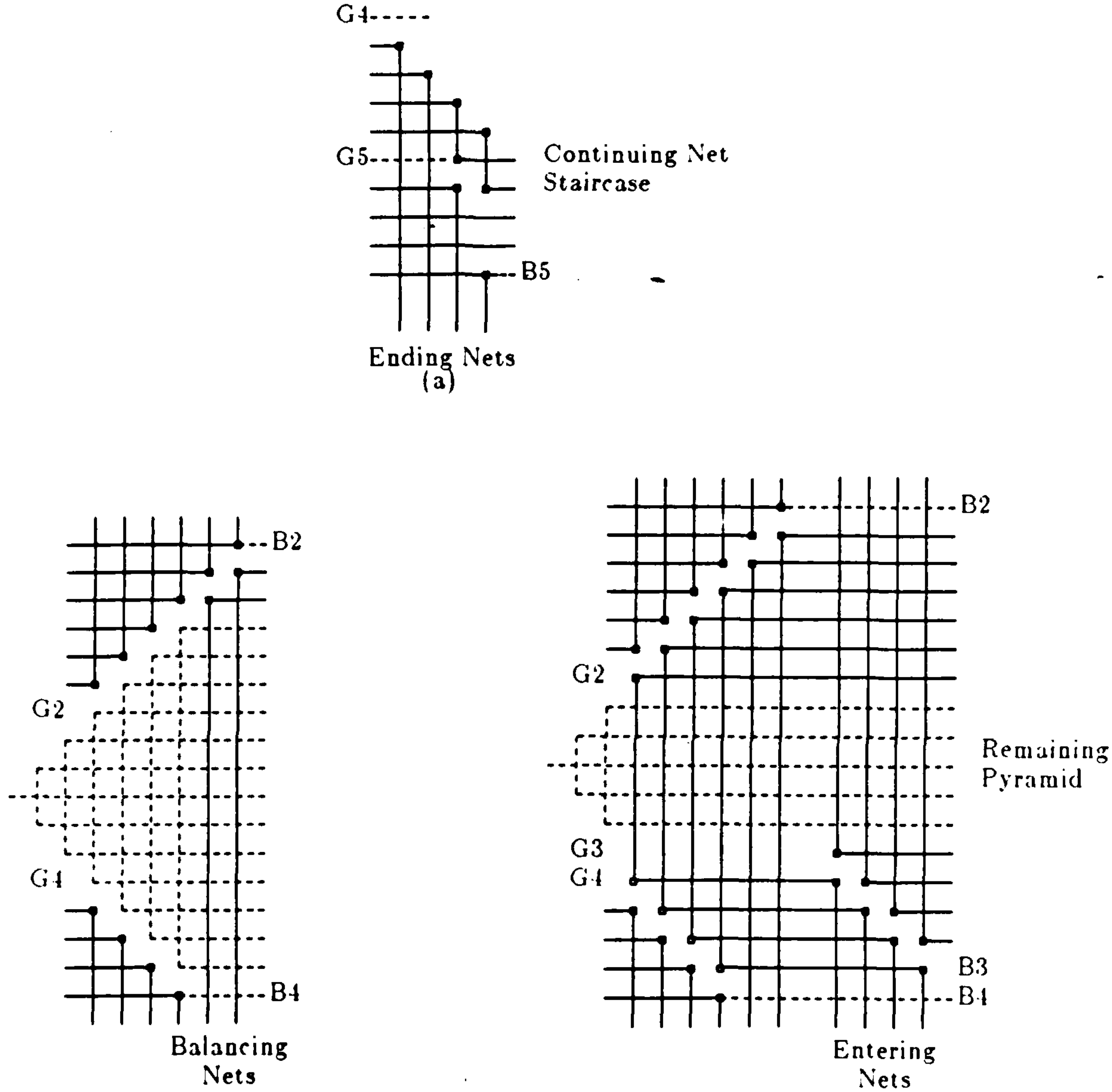


Figure 1. The two phase partition of the unit-vertical-overlap algorithm.



(b)

(c)

Figure 2. Manhattan interblock routing procedure.

(a) Exit all ending nets.
Repack rising/falling structure with continuing nets.

(b) Expand the backtracking pyramid.
Balance excess exiting nets using entering nets.

(c) Route the starting nets.

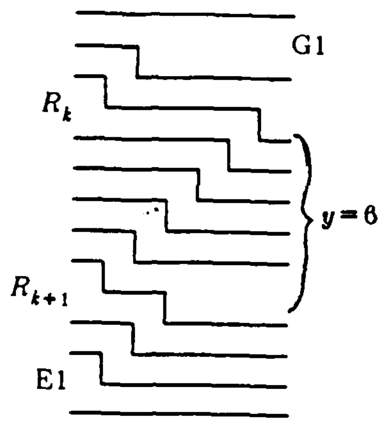


Figure 3. Propagating an extra track E1 to a next block good position G1. R_i denotes reserved track i.

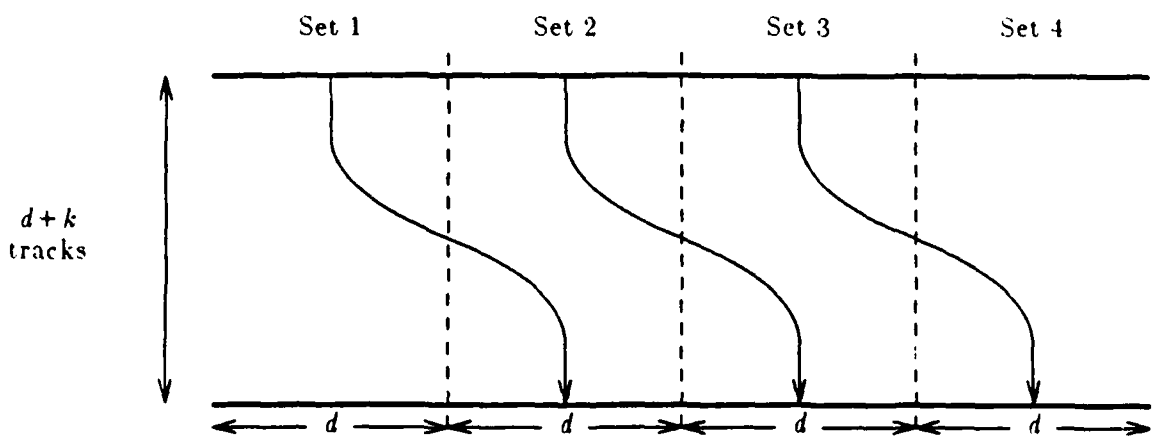


Figure 4. CRP for lower bound argument

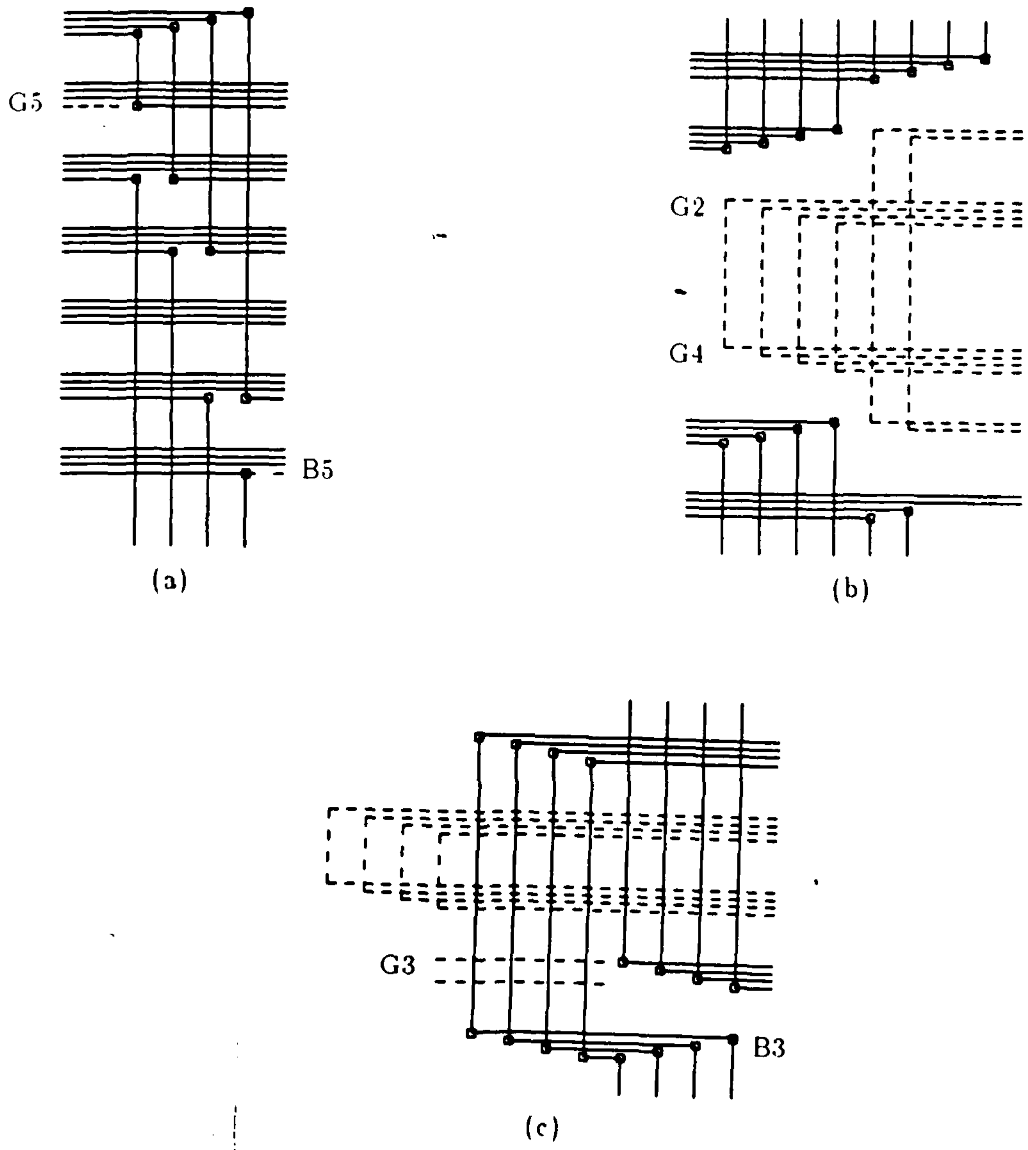


Figure 6. Manhattan interblock routing in L layers.

(a) Exit all ending nets (from layer 2).
Repack rising/falling structure with continuing nets (layers 2 through L).

(b) Expand the backtracking pryamid.
Balance excess exiting nets using entering nets (not pictured).

(c) Route the starting nets, in $L-1$ layers of the pyramid.

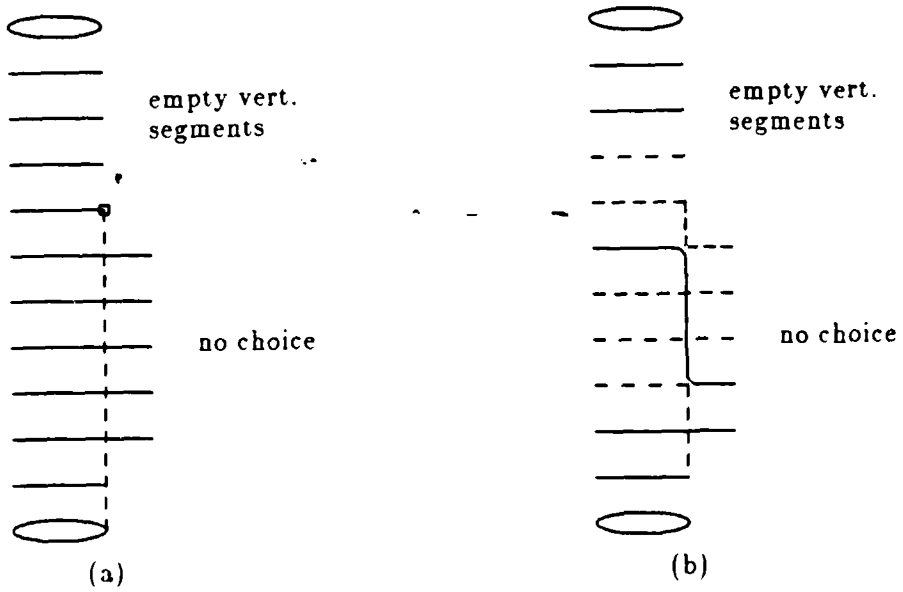


Figure 5. Restricted routing within a group.

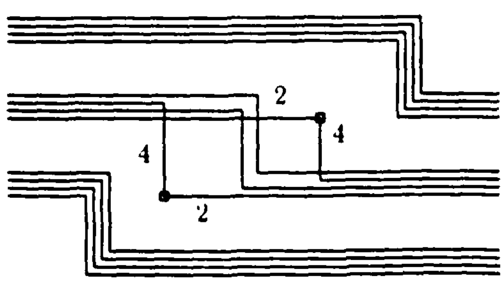


Figure 7. A net in layer 4 is swapped into layer 2 in a (three column) jog.

END

3-87

DTIC